高层建筑安全疏散问题的研究
及泛函背景下疏散方案设计的尝试

杭州外国语学校
周大桐 王雨菡
指导教师 潘俊

2013 年 8 月

摘要

本文运用函数对高层建筑每段楼梯的疏散速率（单位时间内人通量）进行描述，利用泛函等相关知识和计算机软件，给出在指定时间内逃生人数、逃生效率与各楼层分配人数间的关系，同时讨论疏散方式中风险带来的扰动对疏散效率的影响。通过建立这样的模型，试图对高层建筑中人员、店面等的分布安排起一定的指导作用。

Abstract

This article tries to describe the evacuation efficiency (people flowrate per unit of time) in every flight of stairs in a multi-storeyed building, using functional knowledge and computer software, and analyze the relationship between evacuation efficiency and personnel distribution arrangement. At the same time, this article studies how the disturbance brought about by the risk in the evacuation process influences the quantity of successfully evacuated people in a limited period. Building this model, we expect that it can function as, to a certain degree, the guidance of personnel distribution arrangement in a multi-storeyed building.

目录

摘要 2

目录 3

第一章 前言 4

第二章 相关知识准备 5

2.1 偏导数 5

2.2 常微分方程与偏微分方程 5

2.3 变分法 5

2.4 线性代数 5

第三章 建立模型 6

3.1 对楼梯上疏散状况的函数刻画 6

3.1.1 函数提出过程 6

3.1.2 实际测试 7

3.1.3 与学界其他结果的比较 7

3.2 条件弱化模型 8

3.2.1 建立模型 8

3.2.2 数学方法的证明 9

3.2.3 在本模型中的应用 9

3.3 条件强化模型 12

3.3.1 某一瞬间变化对常微分方程的影响 13

3.3.2 连续二阶小量对常微分方程的影响 15

3.3.3 建模过程 17

3.3.4 数学处理以及在 matlab 平台的实现 18

总结 25

致谢 25

参考资料 25

附录 26

第一章 前言

从 2008 年四川汶川大地震，2010 年舟曲泥石流、上海胶州路特大火灾，到今年雅安地震，突发性灾害事件频发，带来惨重的伤亡。各地纷纷加强高密度场所如医院、学校等地的应急疏散演练工作，建筑物紧急状态下的稳固性也作为基础设施建设时的一个指标更多地为建筑者所重视。

但我们发现，为了减轻突发灾害造成的人员伤亡，学界往往致力于在以下两方面进行优化：加强建筑稳固性和优化疏散方案。因此，我们试图提出一种新的思路来更好地应对潜在的危险，即通过调整各楼层的人员分配来加快疏散效率和增强疏散过程中的稳定性。

在实际生活中，高层建筑往往兼有办公、商贸、餐饮等多种功能，而各楼层的功能直接影响了各楼层正常状态下的人口密度，因此我们可以通过调整各楼层的功能来改变人员比例。以学校为例，可以通过改变常用教室、多媒体教室、空教室等在各楼层的分布调整人员分布，而大型商城则可以通过调整餐饮区、购物区、仓库等的位置实现调整。这样的调整对建筑的功能和经济效益影响不大，但会较大地改变紧急疏散的效率和安全性。

确认了该思路的现实可执行性后，我们着手建立模型，由最简单的三层建筑入手（第一层视为可以不经楼梯直接逃生），尝试给出描述各楼梯口疏散效率的方程。在建立方程的过程中，我们需要应用到人员平均速度与已有人员数量的关系，查阅相关文献，我们发现这个方程被通常地表示为负指数函数的形式。但是经过实地测算，我们感到该方程并不适用于我们的模型，因此对其进行了优化改良，采用更合理的曲线描述人员平均速度与已有人员数量的关系，建立了方程组。

运用数学工具分析，我们得到了人员分配比对总体疏散效率的影响，并能够寻找到实现最佳疏散效率的人员分配比。同时，考虑到楼梯口处大量人员进入对原有疏散秩序可能造成的“冲击”效果，我们将其用相应位置的扰动进行描述，观察分析其对最终疏散效果的影响，从而得到人员分配比与疏散风险性的关联。

综合考虑疏散的效率与安全性，我们可以通过调整两者的重要性权重，来找到最佳的人员分配比，同时满足两方面的要求，从而指导建筑的内部布局，实现对突发灾害时人员安全性的进一步保障。

第二章 相关知识准备

2.1 偏导数

一个多变量的函数的偏导数是它关于其中一个变量的导数，而保持其他变量恒定（相对于全导数，在其中所有变量都允许变化）。函数f关于变量x的偏导数写为f'_x或$\frac{\partial f}{\partial x}$。偏导数符号是圆体字母$\partial$，区别于全导数符号的正体$d$

2.2 常微分方程与偏微分方程

常微分方程（Ordinary Differential Equation，简称 ODE）是未知函数只含有一个自变量的微分方程。

偏微分方程（Partial Differential Equation，简称 PDE）指含有未知函数及其偏导数的方程。描述自变量、未知函数及其偏导数之间的关系。符合这个关系的函数是方程的解。偏微分方程分为线性偏微分方程式与非线性偏微分方程式，常常有几个解而且涉及额外的边界条件。

2.3 变分法

变分法是处理泛函的数学领域，和处理函数的普通微积分相对。譬如，这样的泛函可以通过未知函数的积分和它的导数来构造。变分法最终寻求的是极值函数：它们使得泛函取得极大或极小值。

2.4 线性代数

线性（linear）指量与量之间按比例、成直线的关系，在数学上可以理解为一阶导数为常数的函数。非线性（non-linear）则指不按比例、不成直线的关系，一阶导数不为常数。

线性代数起源于对二维和三维直角坐标系的研究。在这里，一个向量是一个有方向的线段，由长度和方向同时表示。现代线性代数已经扩展到研究任意或无限维空间。一个维数为 n 的向量空间叫做 n 维空间。在二维和三维空间中大多数有用的结论可以扩展到这些高维空间。

第三章 建立模型

3.1 对楼梯上疏散状况的函数刻画

3.1.1 函数提出过程

在通道不太长以至于涌入人员对通道流量的滞后效应不明显时，可以视通道流量与人员密度成确定的函数关系。很明显，这是一个先增后减的函数，函数值在正半轴上恒大于 0，且人数为 0 时的流量为 0。

对拥挤状态下人员移动速度与人流密度的研究主要是运用现场观测和录像记录两种手段进行，到目前已经积累了大量的观测数据。其中，具有代表性的研究者有前苏联的 Predtechenskii Milinskii，美国的 Fruin，Maclennan，Nelson, 英国的 Smith，日本的 Ando，加拿大的 Paul 等人。各位学者得到的人流密度与行进速度关系存在较大差异,他们将人流密度与行进速度的关系拟合成线性关系、对数关系、指数关系、三角函数关系等。[1]

由于要用变分原理求泛函极值，且在打靶法求边值问题过程中函数猜测值可能会超过函数在实际应用中的定义域，在初步建模中为了回避计算上不必要的麻烦，倾向于使用能延拓到$\mathbb{R}$上并在$\mathbb{R}$上解析的函数拟合速度与人流密度的关系。同时，为了防止计算时提出的最优解部分超出实际应用中的定义域，应有$\rho \in (-\infty, 0), v(\rho) < 0$，如人流密度与行进速度成负指数关系。但根据在本校出操时对楼梯口通行状况的观察，这个函数在ρ较大时存在一定失真。在人员较多时，楼梯仍然有一个较小的流通速度，而不是像预测的那样快速趋向于 0。在本校楼梯上，这个速度大约为0.61人$/s$。

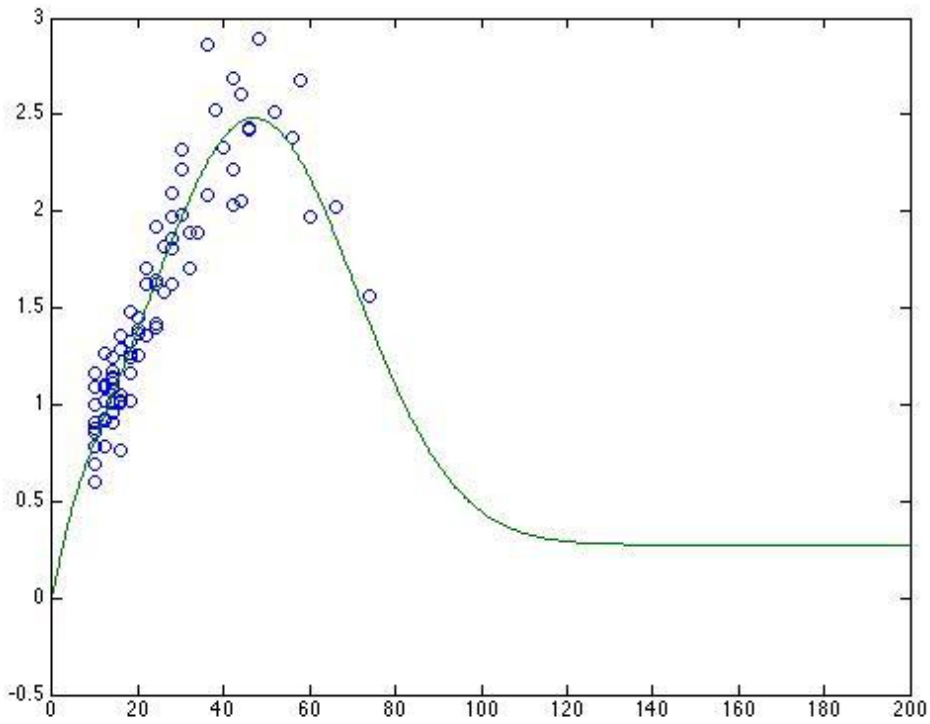
由此，我们想到尝试用截尾正态分布的方法拟合：设$s(x)$为待求函数，

$$s(x) = v_0 + v_1 e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2} - \left(v_0 + v_1 e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2 x_1}{S}}$$

其中，S是通道面积，x_1为人数，其余五个参数要通过试验结果确定。第二项是罚函数，来保证人数为 0 时的流量为 0。而且，第二项的存在可以使$\rho \in (-\infty, 0), v(\rho) < 0$，保证最优化解不会超过函数在实际应用中的定义域。

3.1.2 实际测试

在对本校楼梯的实际测算中，得到楼梯上人数与流量的关系如下：



由于无法发动全校师生为我们做这个试验，在人数较多时的取样较少，而人数较少时则获得了较多样本。
拟合的结果为：
k_1=0.6519
k_2=5.1020
ρ_1=1.760
s=26.77（由测量得到）
v_0=0.2752
v_1=2.2079

3.1.3 与学界其他结果的比较

Fruin 统计得出下楼梯的最佳条件为:密度为2.0人$/m^2$,楼梯有效宽度的流量为1.18人$/s$ [2]

中国科学技术大学火灾科学国家重点实验室采用数码摄像的方法计算得的不同条件下的人流密度、人流量、人流速度参数值显示密度为1.8人$/m^2$时得楼梯的最大流量。[3]

在人数较多时第二项的影响可以忽略，因此：

$$v_{max} \approx v_0 + v_1 = 5.537$$

$$\rho|_{v=v_{max}} \approx \rho_1 = 1.760$$

与以上两结果较为接近。

若在人数不过多时以负指数函数形式拟合速度与人流密度关系，同时以$v = \rho lu = \rho l C_1 e^{-C_2\rho}$形式估计人流量，则达到极限人流量时，每个人平均速度与单个人极限速度比满足$1: e$的定值，与C_1, C_2无关。在本拟合中，这一结果约为$1: 2.5$。

3.2 条件弱化模型

3.2.1 建立模型

规定对任一建筑物均存在一特定时间值t_1，称为安全时间，在紧急情况发生后的t_1时间内可以保证建筑物的安全性，此后安全性不能保证，因此我们的目的就是在t_1时间内实现尽可能多的疏散。

先假设我们只能控制每一层的人数，不能控制单位时间人进入楼道的数量。例如单位时间进入通道的人数与时间成负指数函数关系。对于每一段楼梯，都可以对应一个方程描述其中人数的变化。一般的，设人数为x时的疏散速率为$s(x)$，单位时间从该层进入楼梯的人数随时间变化的函数为$y_i(t)$，则这个方程的形式为：

$$\frac{dx_i}{dt} = -s(x_i) + s(x_{i-1}) + y_i(t)$$

最简单的情况下，可以认为疏散效率为 $s(x) = v_0 + v_1 e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}$。因此，以最简单的三层为例，将一楼看作无需经过楼梯可直接逃离，就可以得到一个方程组：

$$\frac{dx_1}{dt} = f_1 = -s(x_1) + \alpha e^{-mt}$$

$$\frac{dx_2}{dt} = f_2 = -s(x_2) + s(x_1) + (A - \alpha)e^{-mt}$$

$$\frac{dx_3}{dt} = f_3 = s(x_2)$$

其中x_3表示已经从通道出来的人数。A由总人数决定，而α由分配方案决定，因此我们可以通过控制α来找到最优方案。

3.2.2 数学方法的证明

对于一个初值固定的n元一阶常微分方程组，含一个确定的参数α，

$$\frac{dx_i}{dt} = f_i(x_1, \dots, x_n, t, \alpha) \;\; i = 1,2 \dots n$$

$$x_i(0) = x_i|_{t=0}$$

其图像为$n+1$维线性空间中的一条曲线。若引入第$n+2$维α，则对于一个确定的α将有一条这样的曲线，所有这些曲线构成一个$n+2$维空间中的曲面。如果将α也视作变量,且将之前的所有对t求导改为对t求偏导，则形成边界条件为在$t=0$时所有初值已知的偏微分方程：

$$\frac{\partial x_i(t,\alpha)}{\partial t} = f_i(x_1 \dots x_n, t, \alpha) \;\; i = 1,2 \dots n$$

$$x_i(0,\alpha) = x_i|_{t=0}$$

对于确定的方程组，其解是唯一的。

形式地对其求对α的一阶偏导，得到n个方程：

$$\frac{\partial^2 x_i(t,\alpha)}{\partial t \partial \alpha} = f_i(x_1 \dots x_n, t, \alpha) \;\; i = 1,2 \dots n$$

由于α不影响初值，所以$\frac{\partial x_i(0,\alpha)}{\partial \alpha} \equiv 0$

由此可以构造n个辅助方程：

$$\frac{d}{dt}\frac{\partial x_i(t,\alpha)}{\partial \alpha} = f_i(x_1 \dots x_n, t, \alpha) \;\; i = 1,2 \dots n$$

$$\frac{\partial x_i(0,\alpha)}{\partial \alpha} = 0$$

其中任意时刻的$\frac{\partial x_i(0,\alpha)}{\partial \alpha}$代表此时在$n+2$维空间中，曲面上点在$\alpha$方向的斜率（偏导数）。从函数的角度看，对于每一个组(t,α)，都有一个确定的值，因此可以将系统看作对(t,α)的二元函数，对于确定的时间t_x求某一个x_i极值的问题就转化为求使偏导数$\frac{\partial x_i(t_x,\alpha)}{\partial \alpha} = 0$的问题，这是一个两点边值问题，由此可以求出合适的$\alpha$以及$x_i$极值。

3.2.3 在本模型中的应用

根据偏微分基本性质有：

$$\frac{d\frac{\partial x_1}{\partial \alpha}}{dt} = \frac{\partial f_1}{\partial \alpha}$$

$$\frac{d\frac{\partial x_2}{\partial \alpha}}{dt} = \frac{\partial f_2}{\partial \alpha}$$

$$\frac{d\frac{\partial x_3}{\partial \alpha}}{dt} = \frac{\partial f_3}{\partial \alpha}$$

$$\forall \frac{\partial x_i}{\partial \alpha}(0) = 0; \frac{\partial x_3}{\partial \alpha}(t_0) = 0$$

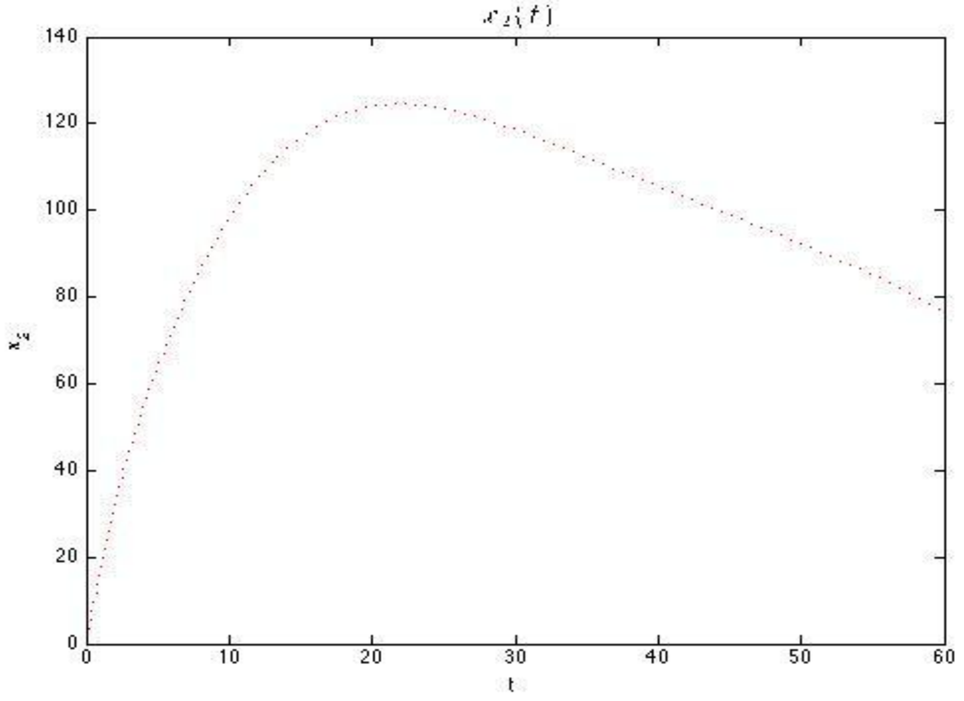
这需要运用到常微分方程的数值运算和牛顿迭代$\alpha_i = \alpha_{i-1} - \frac{\frac{\partial x_3}{\partial \alpha}}{\frac{\partial^2 x_3}{\partial \alpha^2}}$。如果要对$\frac{\partial^2 x_3}{\partial \alpha^2}$进行计算，就需要再形式求偏微分一次，以求得计算$\frac{\partial^2 x_i}{\partial \alpha^2}$。如果考虑更高层的问题，就会出现更多与分配有关的变量，此时需要利用方程组的雅可比矩阵进行牛顿迭代：

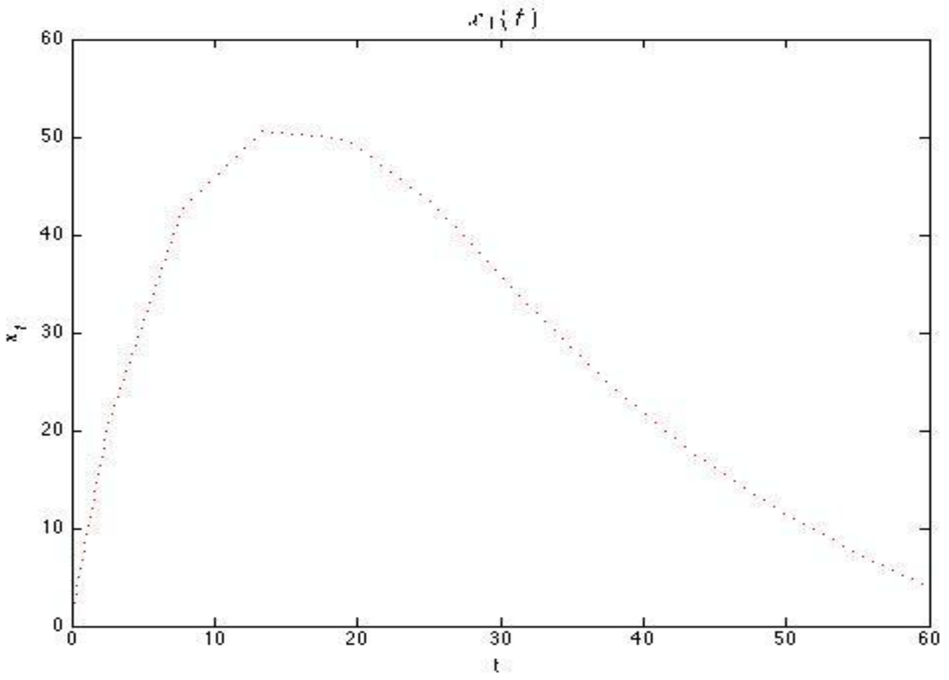
$$A_i = A_{i-1} - J^{-1}(E) \times E$$

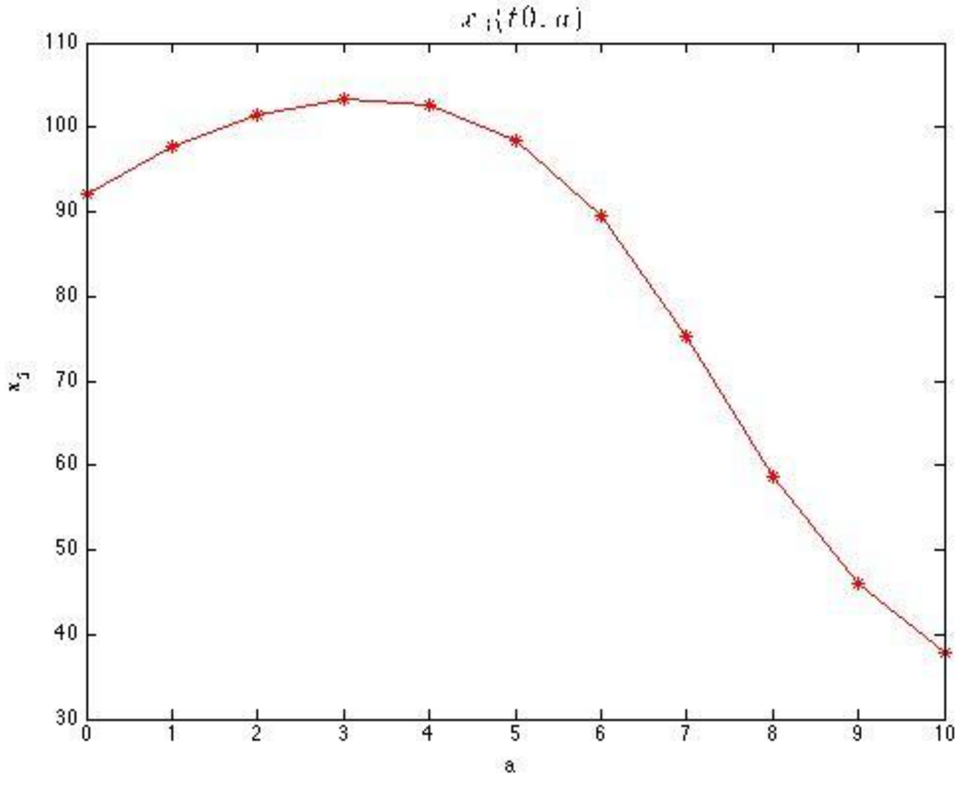
有了这个工具，这个问题已经基本上得到了解决，借助计算机软件可以得到足够好的$s(x)$，计算可以选用打靶法或分割法求边值解。

在本校去年的地震逃生演习中的观察表明，我校教学楼四个楼梯（排布为矩形四角）中的任一个均有$A \approx 10; m \approx 0.09$。当时高三部分班级由于在进行高考复习没有参加演习，因此二、三楼实际参与人数约为 450 人，与按负指数拟合积分的结果大致吻合。

假设我们可以自由调节二、三楼人数，目标时间选取为$t_0 = 60s$，则通过以上计算获得结果如下：







我们可以用以上方法寻找α，并对取极值时x_1, x_2对时间的函数做出精确的计算。但是要画一张精确的$x_3(t_0, \alpha)$相图的计算量很大，因为这意味着要对于差分的每一个α值求常微分方程的数值解并绘图。此处仅以步长为1差分作图，给出较为直观的印象。图中$\alpha = 0$代表所有人分布在 2 楼，$\alpha = 10$代表所有人分布在三楼。以上计算代码见求解代码①

对于三层以上的建筑，同样可以采用这一方法建模并得到各楼层人数之间的

建议比值，唯一的不同在于方程数目以及未知数个数。

如果要刻画人员分布可变时的疏散状况，在之前模型中有确定函数形式的$y(t)$要在新模型中改为只有确定边界值的未知函数。如果我们求出了一个最优疏散方案所要求的$y(t)$来表示单位时间内进入楼梯人数与时间的函数关系，我们就可以在楼层中标定人员跑到楼梯口所需时间$t \in [t_1, t_2]$的对应区间，这一区间在无险情时的人员密度应尽可能接近$\int_{t_1}^{t_2} y(t)dt$。

通过以上数学模型与数学方法，我们已经可以得到对某建筑人员分布有一定指导意义的结论。在实际中，我们可以在建筑设计与空间分配时有意识的预控每一处的人员密度，应用此结论。

在学校中，由于每个行政班的人数大致相同，这一模型的指导意义不如之前的初步模型，但是在其他公共场合做这样的预控还是可能的。以商场为例，各店面附近人员密度相差很大，在估计了每一个楼梯所需要管辖的区间与这一区域的大致总人数后，可以鼓励客流量不同的商家选择与消防要求相匹配的店面。

以下尝试在$y(t)$可变的较强条件下，求出最优的$y(t)$。由于会涉及变分法的边界条件，以下$y(t)$代表某一时刻已经进入楼梯的人数，单位时间内进入楼梯人数与时间的函数关系用$y'(t)$表示。

3.3 条件强化模型

现在进一步考虑在控制每一层人数的同时，我们也能控制单位时间内人进入楼道的数量。那么这个问题就变得复杂了，需要分多步进行处理。对于一般的问题，以三层建筑为例，疏散过程可以粗略地用这样的方程组来描述：

$$\frac{dx_1}{dt} = -s(x_1) + {}_1(t)$$

$$\frac{dx_2}{dt} = -s(x_2) + s(x_1) + y_2(t)$$

$$\frac{dx_3}{dt} = s(x_2)$$

此模型可以给我们一个相对直观的感觉：对于每一个$y_1(t)$、$y_2(t)$，在t_1时刻都有一个确定的$x_3(t_0) = \int_0^{t_0} s(x_2)dt$与之对应。我们可以定义一个$\frac{dx_3}{dt} = s(x_2)$，初值为 0，来代替积分形式。但无论哪种形式，都建立了一个$[y_1(t), y_2(t)] \mapsto x_3$的映射，由泛函定义可知这是二元泛函，我们要求解的正是泛函极值。

这一类中问题中较为简单的问题如：

定义映射$y(t) \mapsto x(t_0)$

$$\frac{dx}{dt} = f(x, y, y', t)$$

$$x(0) = x_0; y(0) = y_0; y(t_1) = y_1$$

求$y(t)$使$x(t_0)$在其附近取极值。

为了解决这个问题，必须进行以下几步考虑：

3.3.1 某一瞬间变化对常微分方程的影响

不妨先考虑初始时刻对初值的微扰对常微分方程的影响。定义函数$f(x_0,t_0)$

$$\frac{dx}{dt}=f(x,t)$$

$$x(0)=x_0$$

$$f(x_0,t_0)=x(t_0)$$

求：

$$\lim_{\Delta\alpha\to 0}\frac{f(x_0+\Delta\alpha,t_0)-f(x_0,t_0)}{\Delta\alpha}$$

为此先讨论对于和x等阶的扰动α所造成的影响：

令$x=x'+\alpha$，原方程可写为

$$\frac{d(x'+\alpha)}{dt}=f(x'+\alpha,t)$$

其中x初值恒为x_0。与之前讨论相同，两边求偏导：

$$左式=\frac{d}{dt}\frac{\partial(x'+\alpha)}{\partial\alpha}=\frac{d}{dt}\frac{\partial x'}{\partial\alpha}$$

$$右式=\frac{\partial f(x'+\alpha,t)}{\partial x'+\alpha}\cdot\frac{\partial(x'+\alpha)}{\partial\alpha}=\frac{\partial f(x'+\alpha,t)}{\partial(x'+\alpha)}\cdot(\frac{\partial x'}{\partial\alpha}+1)$$

因此有：

$$\frac{d}{dt}\frac{\partial x'}{\partial\alpha}=\frac{\partial f(x'+\alpha,t)}{\partial(x'+\alpha)}\cdot(\frac{\partial x'}{\partial\alpha}+1)$$

结合$f(x_0,t_0)$定义有：

$$\lim_{\Delta\alpha\to 0}\frac{f(x_0+\Delta\alpha,t_0)-f(x_0,t_0)}{\Delta\alpha}=\frac{\partial x}{\partial\alpha}+1$$

由此一阶小扰动$\Delta\alpha$在任意时刻对常微分方程中x'与x的影响可表示为：

$$\Delta x'(t_0)=\frac{\partial x}{\partial\alpha}(t_0)\times\Delta\alpha$$

$$\Delta x(t_0)=\Delta(x'(t_0)+\alpha)=(\frac{\partial x}{\partial\alpha}(t_0)+1)\Delta\alpha$$

定义$\beta=\frac{\partial x}{\partial\alpha}+1$，有：

$$\frac{d\beta}{dt}=\frac{\partial f(x'+\Delta\alpha,t)}{\partial(x'+\Delta\alpha)}\beta=\frac{\partial f(x,t)}{\partial x}\beta$$

$$\beta(t_0)=1$$

不妨把β称为放大因子，则t_1时刻x值的变化量可表示为：

$$\Delta x=\beta\Delta\alpha$$

现在,我们将初始时刻微扰对常微分方程的影响推广到任意时刻微扰对常微分方程的影响。定义函数$F(x_0,t_1,t_2,\alpha)$

$$\frac{du}{dt}=f(u,t)$$

$$u(0)=x_0$$

$$x_1=u(t_1)$$

$$u_1 = x_1 + \alpha$$
$$\frac{dv}{dt} = f(v,t)$$
$$v(t_1) = u_1$$
$$F(x_0, t_1, t_2, \alpha) = v(t_2)$$

求：

$$\lim_{\alpha \to 0} \frac{F(x_0, t_1, t_2, \alpha) - F(x_0, t_1, t_2, 0)}{\alpha}$$

解：根据上一步结果，设

$$\frac{dx}{dt} = f(x,t); x(0) = x_0$$
$$\frac{d\beta}{dt} = \frac{\partial f(x,t)}{\partial x}\beta; \beta(t_0) = 1$$

$\alpha \to 0$时对于

$$\frac{du'}{dt} = f(u',t)$$
$$u'(0) = x_0 + \frac{\alpha}{\beta(t_1)}$$

有：

$$u'(t_1) = x_1 + \alpha$$

由唯一性定理：

$$u' \equiv v$$

于是有：

$$v(t_2) - u(t_2) = u'(t_2) - u(t_2) = \frac{\beta(t_2)}{\beta(t_1)}\alpha$$

对于$\frac{dx}{dt} = f(x,t)$,在初值$x(t_0) = x_0 + \Delta\alpha$，其中$\Delta\alpha$为一阶小量时，这一偏差是线性的，所以对于$t_x$时刻的一个大小为$\Delta\alpha$的微扰，在$t_1$时刻被放大为$\frac{\beta(t_1)}{\beta(t_x)}\Delta\alpha$，由此可以由一个放大函数$\beta(t)$得出任意时刻微扰对最终值的影响。

在方程组中计算这一结果：

$$\frac{dx}{dt} = f(x,y,t)$$
$$\frac{dy}{dt} = g(x,y,t)$$

仿照前面的讨论，设在t_0时刻的两个方向的微扰是$\Delta\alpha_x, \Delta\alpha_y$，有：

$$\frac{d\beta_x}{dt} = \frac{\partial f}{\partial x}\beta_x + \frac{\partial f}{\partial y}\gamma_{xy}$$
$$\frac{d\gamma_{xy}}{dt} = \frac{\partial g}{\partial x}\beta_x + \frac{\partial g}{\partial y}\gamma_{xy}$$
$$\frac{d\gamma_{yx}}{dt} = \frac{\partial f}{\partial x}\gamma_{yx} + \frac{\partial f}{\partial y}\beta_y$$

$$\frac{d\beta_y}{dt} = \frac{\partial f}{\partial x}\gamma_{yx} + \frac{\partial f}{\partial y}\beta_y$$

其中初值$\beta = 1, \gamma = 0$。两方向上的扰动：

$$\Delta x = \beta_x \Delta\alpha_x + \gamma_{yx}\Delta\alpha_y$$

$$\Delta y = \gamma_{xy}\Delta\alpha_x + \beta_y\Delta\alpha_y$$

同样，也可以反过来对确定的Δx或Δy计算对应的$\Delta\alpha_x$，$\Delta\alpha_y$。这说明对于任意时刻的一个微扰，都可以和一个确定的初值Δx，Δy等效。令$\gamma_{ii} = \beta_i$，用线性代数的语言，就是：

$$\Gamma = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{n1} \\ \vdots & \ddots & \vdots \\ \gamma_{1n} & \cdots & \gamma_{nn} \end{pmatrix}$$

其中γ_{ij}代表i方向的微扰在j方向的影响。在初值处的矢量微扰被放大到：

$$\begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1} \\ \vdots & \ddots & \vdots \\ \gamma_{1n} & \cdots & \gamma_{nn} \end{pmatrix} \times \begin{pmatrix} \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_n \end{pmatrix}$$

对于任意时刻的矢量微扰，在t_0时刻被放大到：

$$\Gamma(t_0) \times \Gamma^{-1} \times \begin{pmatrix} \Delta\alpha_1 \\ \vdots \\ \Delta\alpha_n \end{pmatrix}$$

设两个微扰为$\Delta\alpha_1, \Delta\alpha_2$，分别是在不同时间的微扰。不失一般性，设$\Delta\alpha_1$是在$t_0$时刻的微扰，则之后的方程就变为了：

$$\frac{d}{dt}(x - \Delta\alpha_1) = f(x - \Delta\alpha_1, t)$$

对这个方程做与之前相同的等效处理，发现在$\Delta\alpha_1 \to 0$时其仍会退化为

$$\frac{d\beta}{dt} = \frac{\partial f(x,t)}{\partial x}\beta$$

就是说，$\Delta\alpha_1, \Delta\alpha_2$之间的相互影响是二阶小量，或者说它们可以线性叠加。也可以发现，每个微扰的放大只取决于常量部分，而与小量无关。如果之前的所有微扰所造成的影响还是一个一阶小量的话，它对下一个微扰所造成的影响也没有影响。因此，对于一些连续的二阶小微扰，我们可以对它们进行积分。

3.3.2 连续二阶小量对常微分方程的影响

可以注意到，这里的二阶小微扰由 y 的变分提供，积分后得到的是一阶小量，不影响之后的扰动。不妨先假设这个二阶小的微扰是显式，很容易得到：

$$\Delta x = \int_{t_0}^{t_1} \frac{\beta(t_1)}{\beta(t)} \Delta^2 \alpha(t) dt$$

回到$\frac{dx}{dt} = f(x, y, y', t)$,对任意时刻$y(t)$的任意变分，其在$dt$对应的瞬间微扰为：

$$(\frac{\partial f}{\partial}\delta y+\frac{\partial f}{\partial y'}\delta y')dt$$

由于系统中显含x,这一微扰最终被放大到：

$$\frac{\beta(t_1)}{\beta(t)}(\frac{\partial f}{\partial y}\delta y+\frac{\partial f}{\partial y'}\delta y')dt$$

积分为：[4]

$$\int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}(\frac{\partial f}{\partial y}\delta y+\frac{\partial f}{\partial y'}\delta y')dt$$

$$\int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}(\frac{\partial f}{\partial y}\delta y+\frac{\partial f}{\partial y'}\delta y')dt$$

$$=\int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y}\delta y\,dt+\int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y'}\delta y'\,dt$$

$$=\int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y}\delta y\,dt+\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y'}\delta y\Big|_{t_0}^{t_1}+\int_{t_0}^{t_1}\delta y\frac{d}{dt}\left(\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y'}\right)dt$$

$$\because \delta y(t_1)=\delta y(t_2),\because \tfrac{\beta(t_1)}{\beta(t)}\tfrac{\partial f}{\partial y'}\delta y\Big|_{t_0}^{t_1}=0$$

$$\therefore \int_{t_0}^{t_1}\frac{\beta(t_1)}{\beta(t)}(\frac{\partial f}{\partial y}\delta y+\frac{\partial f}{\partial y'}\delta y')dt=\int_{t_0}^{t_1}\left(\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y}-\frac{d}{dt}\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y'}\right)\delta ydt$$

$\because \delta y$任意

$$\therefore \frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y}-\frac{d}{dt}\frac{\beta(t_1)}{\beta(t)}\frac{\partial f}{\partial y'}=0$$

此处$\beta(t_1)$可以约掉，但由于在多变量时转置矩阵Γ是不可约的，所以先写成这样保证形式。整理一下，对于一个确定的y（t），我们有：

$$\frac{dx}{dt}=f(x,y,y',t)$$

$$\frac{d\beta}{dt}=\frac{\partial f(x,y,y't)}{\partial x}\beta$$

$$\frac{dy}{dt}=y'$$

其变分为：

$$\int_{t_0}^{t_1}(\frac{\beta_1}{\beta(t)}\frac{\partial f}{\partial y}+\frac{d}{dt}\frac{\beta_1}{\beta(t)}\frac{\partial f}{\partial y'})\delta ydt$$

当它取极值时，有：

$$\frac{\beta_1}{\beta(t)}\frac{\partial f}{\partial y}+\frac{d}{dt}\frac{\beta_1}{\beta(t)}\frac{\partial f}{\partial y'}=0$$

从中我们可以获得$\frac{dy\prime}{dt}$的显式，约束条件为：

$$x(t_0)=x_0, y(t_0)=y_0, y(t_1)=y_1, \beta(t_0)=1, \beta_1=\beta(t_1)$$

这样就再次形成常微分方程的两点边值问题。

3.3.3 建模过程

由 3.1 讨论知：

$$s(x)=v_0+v_1e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}-\left(v_0+v_1e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2x_1}{S}}$$

作为能控制单位时间内人进入楼道数量的模型，规定时间里疏散出的人数 $\int_0^{t_1} v_0x_4e^{-kx_4}dt$已经不是唯一重要的了，我们所采用疏散方式的风险也要考虑在内。否则我们大可以在$t=0$时刻让一大批人在尽可能短的时间内进入楼梯，以使楼梯的疏散速率达到极大值，而完全不顾这一举动对楼梯口所造成的危险，这就是 3.4 开始的模型符合数学工具的要求但实际上并不合理的原因。

大多数公共场合的楼梯都设有防火门、转角等障碍，即使是与楼层直接相连的楼梯，在进入时人流宽度也会减小一半，因此此处也成为了各种意外的高发地。在定量估计其造成的风险时，可以视风险R_{CT}为

$$R_{CT}=\lambda(N_s/N_t)F_N$$

（取自《人群拥挤踩踏事故后果微观建模及模拟分析》式中N_s为 t 时刻特定区域内的滞留人数，F_N为 t 时刻特定区域拥挤人群中的死亡人数，N_t为设计人数，λ为触发因子，可通过专家打分和评定得到）[5]

在本模型中，囿于计算能力和调查能力，近似设单位时间的涌入人数正比于这段时间内入口附近人数，即

$$y'\propto N_s$$

原文中F_N是由拥挤力达到 242N 持续 3min 时造成人员伤亡判定的，而本模型中假设伤亡以外的因素也会导致判定，因此取消 3min 的延迟时间（这一假设也会带来误差，因为长期拥挤带来的体力下降也可能对意外发生的期望有影响），只认为$F_N\propto N_s$。

因此，任意时刻的风险预期可表示为$\lambda(y')^2$，其中参数待定。对于本校楼梯，可以认为$\lambda=0.04$（这一结果是通过观察本校学生第三节课下课时下楼抢购课间餐的行为进行估计的，不精确之处难以避免，希望在有条件时能得到更精确的结果使得模型的实际意义更强）。这个量对时间积分就是全疏散过程中总风险的期望。

对一个疏散方案作评价时应同时考虑到通道的情况和入口的风险，设x_2为评价函数，则$x_2=\int_0^{t_0} v_0+v_1e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}-\left(v_0+v_1e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2x_1}{S}}-\lambda(y')^2$。写成微分形式，则：

$$\frac{dx_1}{dt}=-v_0-v_1e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}+\left(v_0+v_1e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2x_1}{S}}+y'$$

$$\frac{dx_2}{dt}=v_0+v_1e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}-\left(v_0+v_1e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2x_1}{S}}-\lambda(y')^2$$

$$x_1(0)=x_2(0)=0$$

$$y(0)=0, y(t_0)=y_0$$

求$y(t)$使$x_2(t_0)$取极值。
以下以 s = 10;v0 = 0.182;v1 = 1.46;的小通道，管辖半径内最远的人在 50s 内到达入口，预计疏散 80 人为例计算。

3.3.4 数学处理以及在 matlab 平台的实现

同 3.4 的讨论，对于一个初始时刻的微扰

$$\begin{pmatrix}\Delta\alpha_1\\ \Delta\alpha_2\end{pmatrix}$$

会被放大到

$$\begin{pmatrix}\Delta x_1\\ \Delta x_2\end{pmatrix}=\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}$$

因此在 t 时刻

$$\begin{pmatrix}\Delta\alpha_1(t)\\ \Delta\alpha_2(t)\end{pmatrix}$$

可以等效为$t=0$时刻的

$$\begin{pmatrix}\Delta\alpha_1(0)\\ \Delta\alpha_2(0)\end{pmatrix}=\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\Delta\alpha_1(t)\\ \Delta\alpha_2(t)\end{pmatrix}$$

其变分为：

$$\int_0^{t_1}\begin{pmatrix}\gamma_{11}(t_1) & \gamma_{21}(t_1)\\ \gamma_{12}(t_1) & \gamma_{22}(t_1)\end{pmatrix}\times\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\frac{\partial f_1}{\partial y}\delta y+\frac{\partial f_1}{\partial y'}\delta y'\\ \frac{\partial f_2}{\partial y}\delta y+\frac{\partial f_2}{\partial y'}\delta y'\end{pmatrix}dt$$

$$\int_0^{t_1}\sum_{i=1}^{5}\left(\begin{pmatrix}\gamma_{11}(t_1) & \gamma_{21}(t_1)\\ \gamma_{12}(t_1) & \gamma_{22}(t_1)\end{pmatrix}\times\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\frac{\partial f_1}{\partial y}\\ \frac{\partial f_2}{\partial y}\end{pmatrix}\right.$$

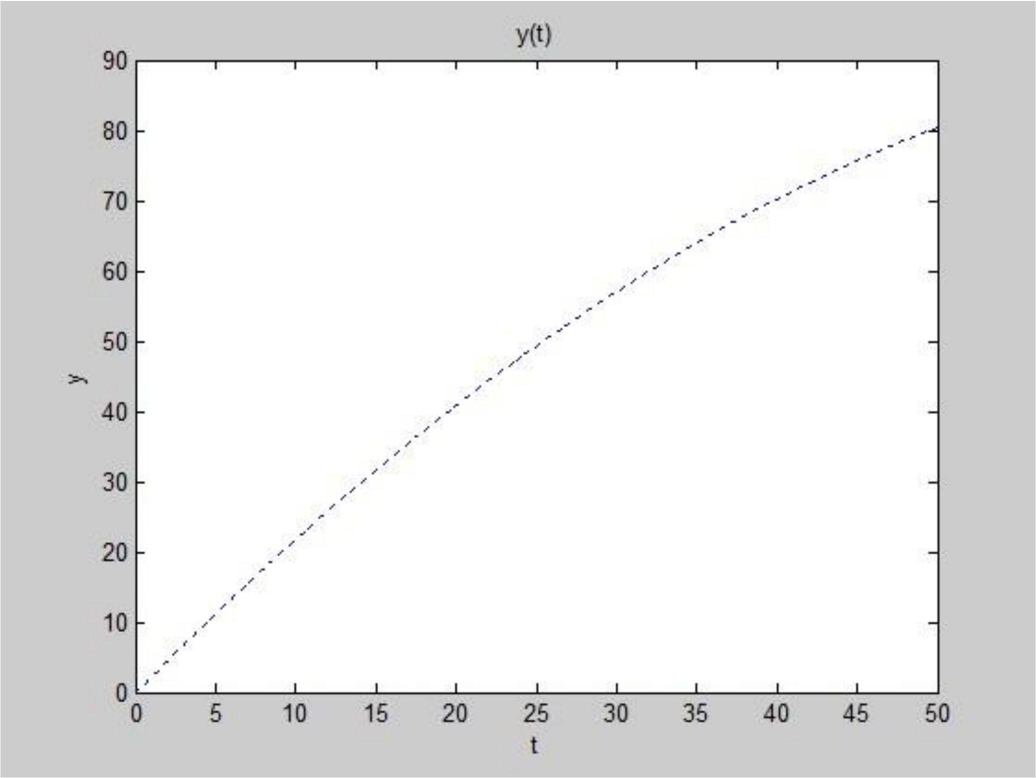
$$\left.-\frac{d}{dt}\left(\begin{pmatrix}\gamma_{11}(t_1) & \gamma_{21}(t_1)\\ \gamma_{12}(t_1) & \gamma_{22}(t_1)\end{pmatrix}\times\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\frac{\partial f_1}{\partial y'}\\ \frac{\partial f_2}{\partial y'}\end{pmatrix}\right)\right)\delta u_i\,dt$$

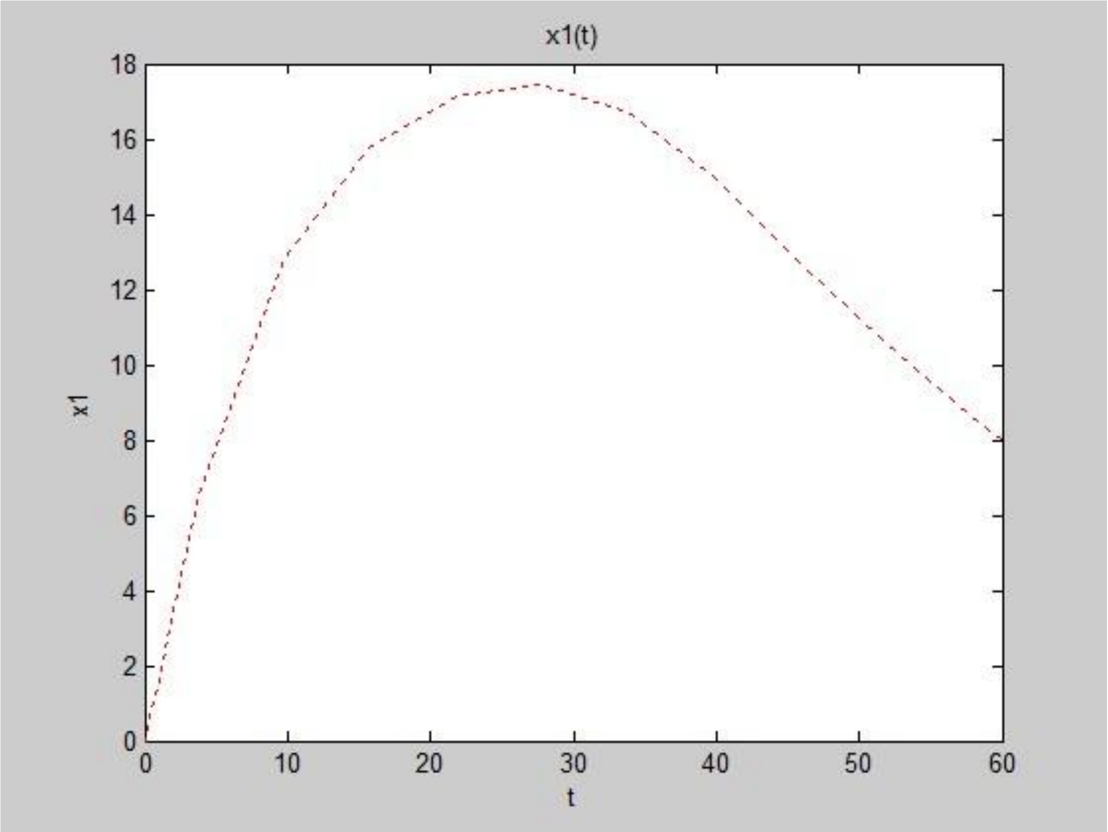
由于δu_i任意性，为使对于任意$x_2(t_1)$取极值，有：

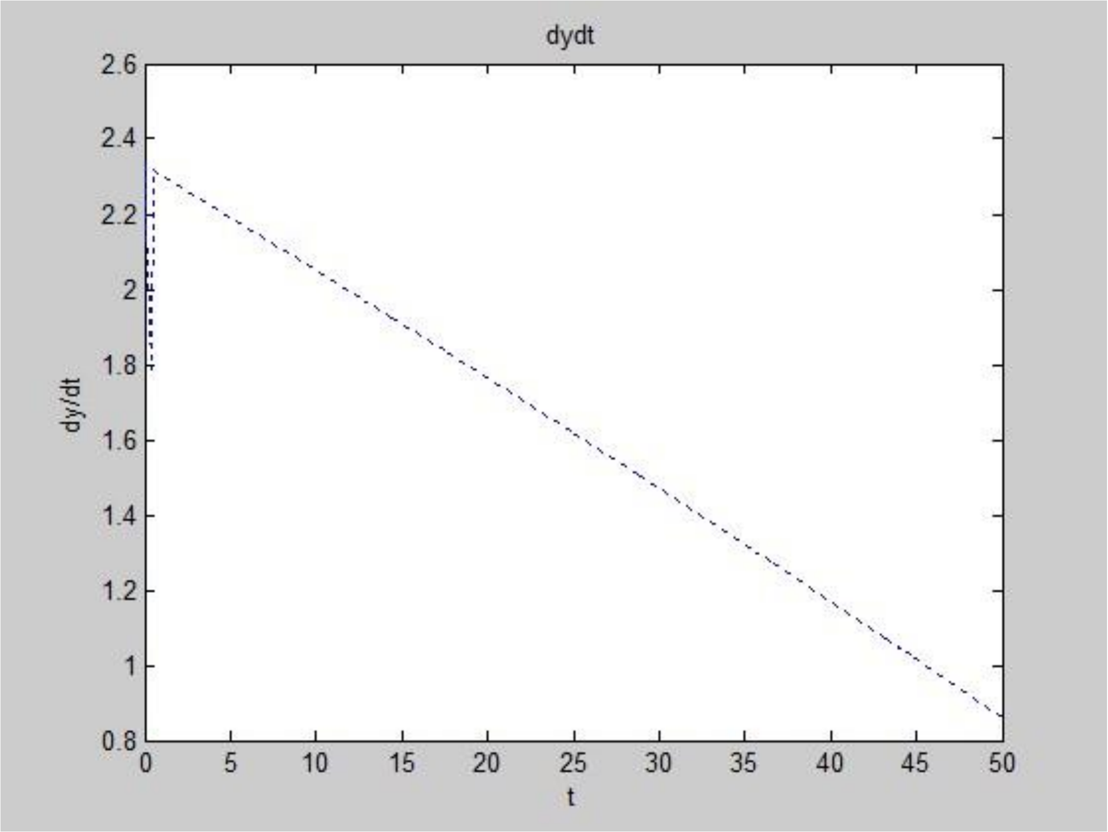
$$\begin{pmatrix}\gamma_{11}(t_1) & \gamma_{21}(t_1)\\ \gamma_{12}(t_1) & \gamma_{22}(t_1)\end{pmatrix}\times\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\dfrac{\partial f_1}{\partial y}\\ \dfrac{\partial f_2}{\partial y}\end{pmatrix} - \frac{d}{dt}\left(\begin{pmatrix}\gamma_{11}(t_1) & \gamma_{21}(t_1)\\ \gamma_{12}(t_1) & \gamma_{22}(t_1)\end{pmatrix}\times\begin{pmatrix}\gamma_{11} & \gamma_{21}\\ \gamma_{12} & \gamma_{22}\end{pmatrix}^{-1}\times\begin{pmatrix}\dfrac{\partial f_1}{\partial y'}\\ \dfrac{\partial f_2}{\partial y'}\end{pmatrix}\right)$$

的第二行为 0

计算出满足边界条件的解如图所示





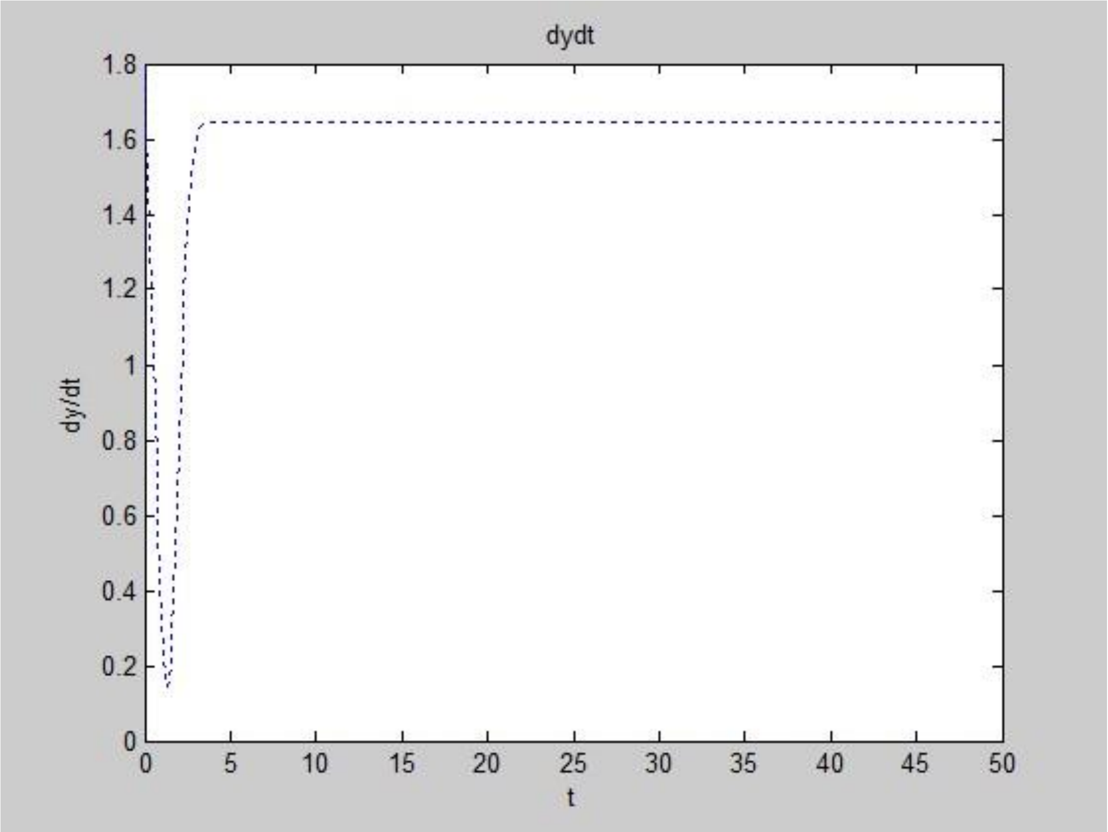


在计算的开始部分图像有大范围震荡，这是由于计算方法引起误差。在修改

初始猜测值时震荡的大小和方向都随之变化，而后面主要的图像不变。其中第三张图像对某一通道所管辖半径内人员的分布有一定指导意义。

和其他以迭代为基础的方法一样，本方法的成功与否很大程度上依赖于初值是否能使迭代向所需方向收敛，而且由于需要猜测的包括转置矩阵的元素，这些元素在$y(t)$未知时较难估计，因此对初值猜测的要求较一般迭代问题要高。

如果初值选取不当，也会出现这种解：



以上计算代码见求解代码②

本校实际疏散情况为有 4 个对称分布在矩形四角的 4 个楼梯。易证优化方案一定满足四个楼梯之间的对称。假设 2、3 楼之间的楼梯压力较小，暂时优先考虑 1、2 楼之间楼梯承受的压力，这样，我们需要求的 1、2 楼之间楼梯单位时间进入人数与时间的关系同样可以由以上方程组定义的泛函极值得到。代入之前测得的数据：

k_1=0.6519

k_2=5.1020

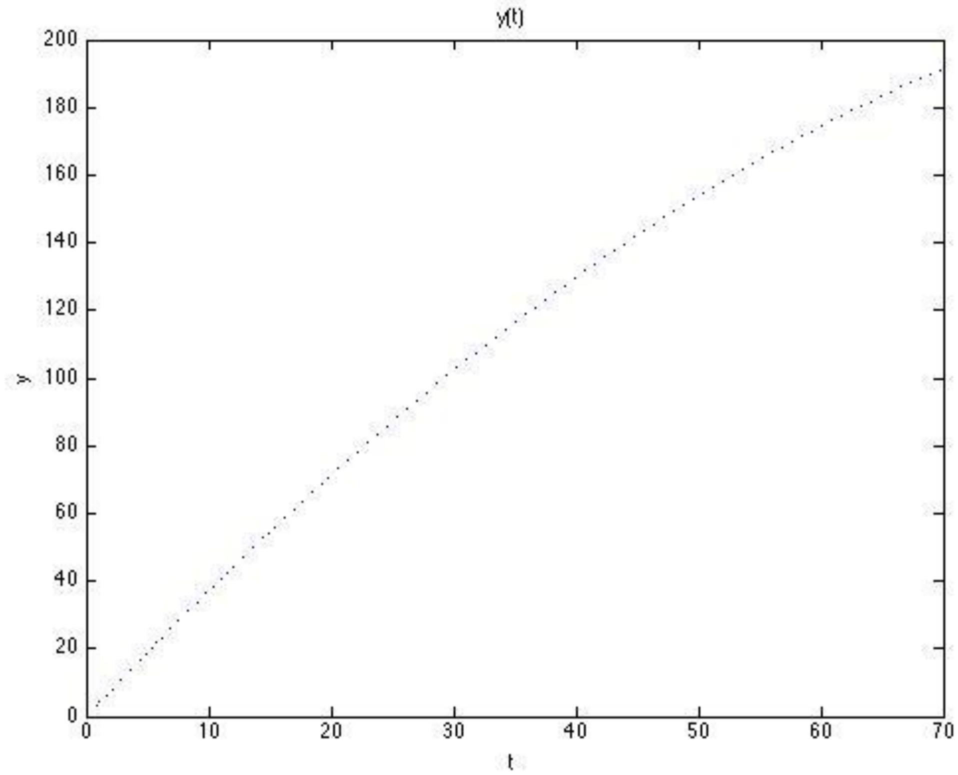
ρ_1=1.760

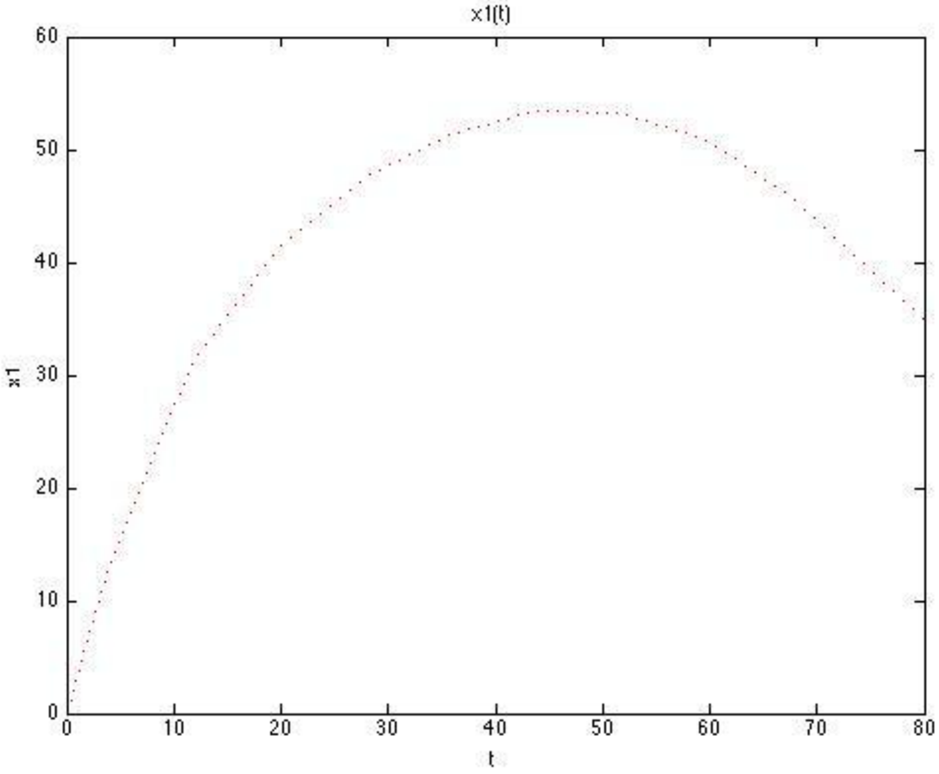
s=26.77（由测量得到）

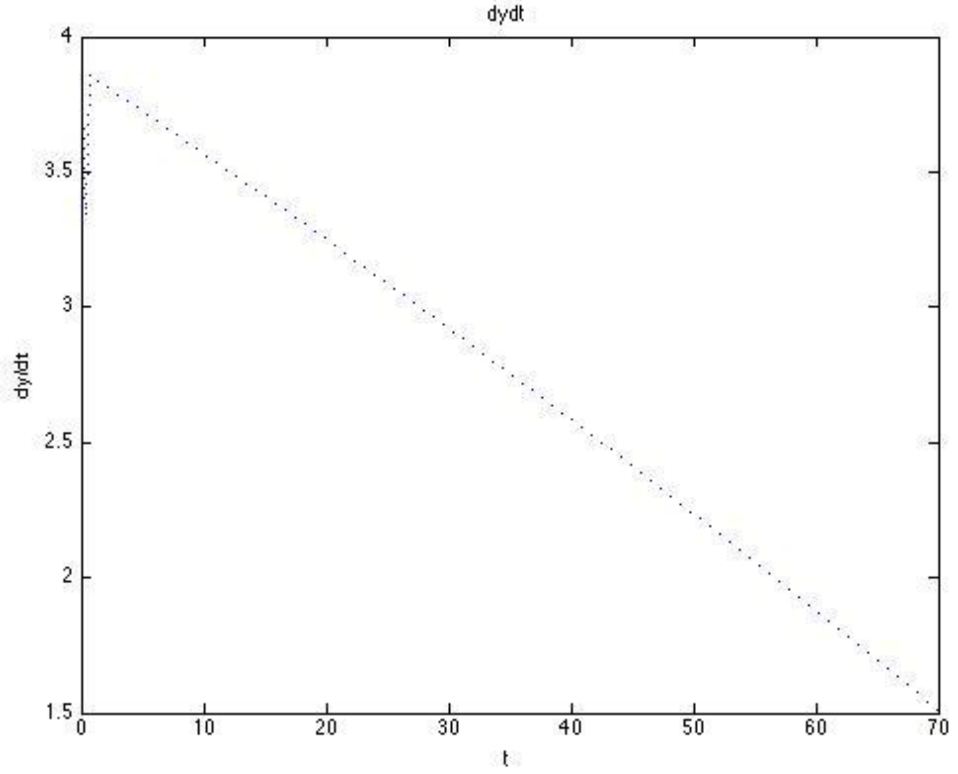
v_0=0.2752

v_1=2.2079

解的图像如下：


y(t)
y
t
200
180
160
140
120
100
80
60
40
20
0
0
10
20
30
40
50
60
70


x1(t)
x1
t
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80



以上计算代码见求解代码③

如果认为 2、3 层间的楼梯所承受压力也要计入，则需要再添加方程。事实上，和 3.2 的模型一样，对于 3 层或以上的疏散同样可以利用变分原理得到解，只需要添加方程与未知数即可，但是在解的过程中要引入泛函极值的变边值方法。如三层问题的所有方程与边界条件为：

$$\frac{dx_1}{dt} = -v_0 - v_1 e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2} + \left(v_0 + v_1 e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2 x_1}{S}} + {y_1}'$$

$$\frac{dx_2}{dt} = -v_0 - v_1 e^{-k_1\left(\frac{x_2}{S}-\rho_1\right)^2} + \left(v_0 + v_1 e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2 x_2}{S}} + v_0 + v_1 e^{-k_1\left(\frac{x_1}{S}-\rho_1\right)^2}$$

$$- \left(v_0 + v_1 e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2 x_1}{S}} + {y_2}'$$

$$\frac{dx_3}{dt} = v_0 + v_1 e^{-k_1\left(\frac{x_2}{S}-\rho_1\right)^2} - \left(v_0 + v_1 e^{-k_1(\rho_1)^2}\right)e^{-\frac{k_2 x_2}{S}} - \lambda(({y_1}')^2 + ({y_2}')^2)$$

$$x_1(0) = x_2(0) = x_3(0) = 0$$

$$y_1(0) = y_2(0) = 0; y_1(t_0) + y_2(t_0) = y_0$$

求y_1, y_2使$x_3(t_0)$取极值。（要写出3×3的转置矩阵Γ的方程，方法如 3.4.1，这里不再展开。）对应的欧拉方程组

$$\Gamma(t_0)\times\Gamma^{-1}(t)\times\begin{pmatrix}\frac{\partial f_1}{\partial y_i}\\ \frac{\partial f_2}{\partial y_i}\\ \frac{\partial f_3}{\partial y_i}\end{pmatrix} - \frac{d}{dt}\left(\Gamma(t_0)\times\Gamma^{-1}(t)\times\begin{pmatrix}\frac{\partial f_1}{\partial {y_i}'}\\ \frac{\partial f_2}{\partial {y_i}'}\\ \frac{\partial f_3}{\partial {y_i}'}\end{pmatrix}\right), i = 1,2$$

的第三行为 0，因此又增加了两个方程，再添加$\frac{dy_i}{dt} = y_i', i = 1,2$以使其成为动力系统，就有 16 个方程，同时转置矩阵的终值共 9 个元素为待定参数。方程中有原始边界条件 6 个，转置矩阵初值等于单位矩阵与转置矩阵终值等于猜测矩阵各贡献 9 个边界条件，共 24 个，还缺少一个。这一个是由泛函极值的变边值方法决定的，不难发现由于限制条件$y_1(t_0) + y_2(t_0) = y_0$，两个端点变分并不是独立的。端点变分

$$\delta F(t_0) = \Gamma(t_0) \times \Gamma^{-1}(t) \begin{pmatrix} \frac{\partial f_1}{\partial {y_1}'} \\ \frac{\partial f_2}{\partial {y_1}'} \\ \frac{\partial f_3}{\partial {y_1}'} \end{pmatrix} \delta y_1(t_0) + \Gamma(t_0) \times \Gamma^{-1}(t) \begin{pmatrix} \frac{\partial f_1}{\partial {y_2}'} \\ \frac{\partial f_2}{\partial {y_2}'} \\ \frac{\partial f_3}{\partial {y_2}'} \end{pmatrix} \delta y_2(t_0)$$

这个矩阵的第三行为 0。

结合$y_1(t_0) + y_2(t_0) = y_0$，得到最后一个边界条件：

$$\Gamma(t_0) \times \Gamma^{-1}(t) \begin{pmatrix} \frac{\partial f_1}{\partial {y_1}'} \\ \frac{\partial f_2}{\partial {y_1}'} \\ \frac{\partial f_3}{\partial {y_1}'} \end{pmatrix} \delta y_1(t_0) - \Gamma(t_0) \times \Gamma^{-1}(t) \begin{pmatrix} \frac{\partial f_1}{\partial {y_2}'} \\ \frac{\partial f_2}{\partial {y_2}'} \\ \frac{\partial f_3}{\partial {y_2}'} \end{pmatrix} \delta y_2(t_0)$$

这个矩阵的第三行为 0。

由此可见，通过变分原理求得优化方案在理论上是可行的。和 3.2 模型一样，在变量和函数增多时差分法所需差分的维度增高，误差将呈几何级数放大，这样的方法可以弥补这一缺陷。虽然计算量较大，但是它比直接差分法寻找最优点（寻找函数时整个函数的定义域都需要差分）或无目的尝试要简便许多。

总结

本文采用分析的方法研究这一模型，在大部分情况下收到了较好的效果，但是和所有采用迭代方法的计算一样，这一方法也依赖于选取初值的收敛性，所幸的是一般的疏散模型中函数的凹凸性都较单一， 否则迭代法的成功率会较小。

本文的模型建立囿于时间和研究条件的限制，采集的数据量不够大，但是对楼梯疏散问题的研究该方法具有普适性，如果能将其与足够精确的预报模型结合，或将使得模型具有更大的实践价值。

致谢

浙江工业大学方兴老师在成稿之初对文中数学处理方法的严谨性进行了检查，浙江大学博士生沈永康先生在计算过程中给予了编程上的指导，杭州外国语学校王敏老师对摘要英文翻译的指正，特此感谢！

参考资料

[1]李三兵，陈峰，李程垒 对地铁站台集散区客流密度与行进速度的关系探讨 城市交通轨道研究 2009 年 12 期

[1.1]Paul, J.L. Effective-Width model for crowd evacuation flow on stairs[A]. Proceedings of 6th International Fire Protection Engineering Seminar[C]. Karlsruhe, Germany, 1982. 295-306.

[1.2]Pauls, J.L. Building Evacuation: Research Findings and Recommendations. Fires and Human Behaviour[M]. New York: John Wiley and Sons, 1980. 251-275.

[1.3]Fruin, JJ. Pedestrian planning and design[M]. New York: Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[1.4]Smith, R.A.Density, velocity and flow relationships for closely packed crowds[J]. Safety Science,1995, 18(4),321-327.

[2]Fruin, JJ. Pedestrian planning and design[M]. New York: Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[3]饶平,杨立中,朱孔金,章涛林 紧急状态下高校典型学生人群楼梯疏散特性研究 中国科学技术大学火灾科学国家重点实验室,合肥 230026

[4]吴崇试 数学物理方法 第二版 北京大学出版社

[5]张青松，刘金兰，赵国敏 人群拥挤踩踏事故后果微观建模及模拟分析 安全与环境学报 2008 年 8 月

附录

求解代码：

①3 层建筑中 2、3 楼人员分布对应疏散状况模型

```
function r = bf3

k1 = 0.6519;
rou1 = 1.7600;
s = 26.7728;
v0 = 0.2752;
v1 = 2.2079;

t0 = 60;

m = 0.09;
A = 10;
lambda = 0:.1:1;
a = A*lambda;

   function x3t0 = ff(alpha)
        %%
        sfun = @(x) v0 + v1.*exp(-k1.*(x./s - rou1).^2);

        figure(1);
        plot(0:100,sfun(0:100),'k:');
        xlabel('$\mathrm{x}$','fontsize',15, 'interpreter','latex');
        ylabel('$\mathrm{s}$','fontsize',15, 'interpreter','latex');

title('$\mathrm{s}{(x)}$','interpreter','latex','fontsize',15);

            function dx1 = Dx1DtSubFun(t, x1)
                dx1 = -sfun(x1) + alpha.*exp(-m*t);
            end

        tspan = 0:60;
        %tspan = [0 60];
        options = odeset('RelTol', 0.0001);
        sol = ode45(@Dx1DtSubFun, tspan, 0, options);

        t45 = sol.x;
```

```
        x45 = sol.y;

        figure(2);clf; h1 = axes; %hold on;
        plot(h1, t45, x45, 'r:');
        xlabel('\itt'); ylabel('\itx_1'); title('$x_1(t)$',
'interpreter', 'latex', 'fontsize', 15);

        h = .1;%²½³¤
        tneed = 0:h:t0;
        [valx1,derx1] = deval(sol, tneed);

        g = derx1 + A*exp(-m*tneed);
        %R-K
        x2 = zeros(length(tneed),1);
        for i = 1:length(tneed)-1
            K1 = h*(-sfun(x2(i)) + g(i));
            K2 = h*(-sfun(x2(i)+ K1/2) + (g(i)+g(i+1))/2 );
            K3 = h*(-sfun(x2(i)+ K2/2) + (g(i)+g(i+1))/2 );
            K4 = h*(-sfun(x2(i)+ K3) + g(i+1));
            x2(i+1) = x2(i) +  1/6*(K1 + 2*K2 + 2*K3 + K4);
        end

        figure(3);
        plot(tneed, x2, 'r:');
        xlabel('\itt'); ylabel('\itx_2'); title('$x_2(t)$',
'interpreter', 'latex', 'fontsize', 15);

        %»ý·ÖÇóx3|t0
        figure(5)
        plot(tneed, sfun(x2));
        x3t0 = h*sum( sfun(x2(1:end-1))+0.5*diff(sfun(x2)) );
    end


x3 = arrayfun(@ff,a);
figure(4);
plot(a, x3,'r*-');
xlabel('\ita'); ylabel('\itx_3'); title('$x_3(t0,a)$',
'interpreter','latex', 'fontsize', 15);
end
```

②条件较强模型的泛函方法在较小通道中的试验

```
function r = bf
lambda = 0.015;
```

```
k1 = 0.6519;
k = 5.1020;
rou1 = 1.76;
s = 26.7728;
v0 = 0.6137;
v1 = 4.9234;

%%²Â²â
t0 = 70;
yt0 = 300;
step = 0.1;%»ý·Ö²½³¤
Num = 3;%µü´ú´ÎÊý

%%³õÖµ;
x10 = 0;
x20 = 0;
y0 = 0;

%%
m1 = 2*k1*v1/s;
m2 = k/s*(v0+v1*exp(-k1*rou1^2));


%Df = df1/dx1£¨Æ«µ¼£©
Df = @(x1)m1.*(x1./s-rou1).*exp(-k1.*(x1./s-rou1).^2) -
m2.*exp(-k./s.*x1);
figure(3);
plot(0:100,Df(0:100),'k:');xlabel('x1'); ylabel('Df');
title('df1/dx1');

%%
   function dx1 = Dx1DtSubFun(t, x1)
      dx1 =
-v0-v1*exp(-k1*(x1/s-rou1).^2)+(v0+v1*exp(-k1*rou1^2))*exp(-k*x1/s);
      dy = zeros(size(t));
      if length(t)>1
         for j = 1:length(t)
            if t(j) < 1
               dy(j) = -(beta(t0)*exp(-t(j)*Df(0.5*t(j))) + c2) /
(2*lambda);
            else
               dy(j) = -(beta(t0) /beta(t(j))+c2)/(2*lambda);
            end
         end
```

```
        else
            if t < 1
                dy = -(beta(t0)*exp(-t*Df(0.5*t)) + c2) / (2*lambda);
            else
%                 if isnan(beta(t)) && ~isnan(beta(t0))
%                     disp(['t:',num2str(t)]);
%                     disp(['beta(t):',num2str(beta(t)),'
beta(t0):',num2str(beta(t0))]);
%                 end
                dy = -(beta(t0) /beta(t)+c2)/(2*lambda);
            end
        end
        dx1 = dx1 + dy;
    end




epsilon = 10^(-3);
beta = @(t)exp(-0.003*t);%²Â²â³õÊ¼Öµ-0.0001 µü´ú5´Î
betafirst = beta;

tspan = [0, t0+10];
x10 = 0;

for kk = 1:2
    for k = 1:Num

    tao = 0:step:t0;
    tmp = 0;
    for i = 1:length(tao)-1
        if tao(i)<1%ÎªÁËÈ¥³ýwarning
            tmp = tmp + step*exp(-tao(i)*Df(0.5*tao(i)));
        else
            tmp = tmp + step*1/beta(tao(i));
        end
    end
    integ = tmp;%»ý·Ö

    c2 = (-beta(t0) * integ - 2*lambda*yt0 )/t0;
    %c2 = c2 - 1

    dy0 = -1/(2*lambda)*(beta(t0)+c2);
    dyt0 = -(1+c2)/(2*lambda);
    dx1t0 = dyt0 - lambda*dyt0^2;
```

```
    options = odeset('RelTol', 0.001);
    sol = ode45(@Dx1DtSubFun, tspan, x10, options);
    t45 = sol.x;
    x45 = sol.y;

    figure(1);clf; h1 = axes; %hold on;
    figure(2);clf; g1 = axes; %hold on;
    plot(h1, t45, x45, 'r:'); xlabel(h1, 't'); ylabel(h1, 'x1'); title(h1,
'x1(t)');
    plot(g1, x45, Dx1DtSubFun(t45, x45), 'b-'); xlabel(g1, 'Î»ÒÆx1');
ylabel(g1, 'ËÙ¶Èdx1'); title(g1, 'x1µÄÏàÆ½Ãæ');



    [x45t0,dx45t0] = deval(sol, t0);

    %%
    dy = @(tt)-(beta(t0)/beta(tt) + c2) / (2*lambda);
    z = 0:step:t0;
    y = zeros(size(z));
    dydt = zeros(size(z));
    for i = 1:length(z)
        if z(i)<.5%ÎªÁËÈ¥³ýwaring
            dydt(i) = -(beta(t0)*exp(-z(i)*Df(0.5*z(i))) + c2) /
(2*lambda);
        else
            dydt(i) = dy(z(i));
        end

        if i==1
            y(1) = 0;
            continue;
        end
        y(i) = y(i-1)+ step*0.5*(dydt(i)+dydt(i-1));%»ý·Ö
    end

    %ÏÔÊ¾y(t)
    figure(4);
    plot(z, y, 'b:');xlabel('t'); ylabel('y'); title('y(t)');
    %ÏÔÊ¾y'(t)
    figure(5);
    plot(z, dydt, 'b:');xlabel('t'); ylabel('dy/dt'); title('dydt');
```

```
    %%
    if kk == 1
        dydtsquare = dydt.^2;
        x2t0tmp = -x45t0 + yt0 - lambda * step * sum(dydtsquare(1:end-1)
+ 0.5*diff(dydtsquare))

        if k == 1
            x2t0 = x2t0tmp;
            bestnum = k;
        end

        if x2t0 < x2t0tmp & x2t0tmp > 0
           x2t0 = x2t0tmp;
           bestnum = k;
        end
    end
    %%
    betatmp = @(t)Df(deval(sol, t));
    beta = @(tt)exp(quadl(betatmp, 0, tt));
    end

    if kk == 1
        Num = bestnum;
        disp(['bestnum:',num2str(bestnum)]);
        beta = betafirst;%²Â²â³õÊ¼Öµ
    end
end



end
```

③条件较强模型的泛函方法在本校教学楼消防中的应用尝试

```
function r = bf2
lambda = 0.04;
k1 = 0.6519;
k = 5.1020;
rou1 = 1.76;
s = 26.7728;
v0 = 0.2752;
v1 = 2.2079;
```

```
%%²Â²â
t0 = 70;
yt0 = 191;
step = 0.1;%»ý·Ö²½³¤
Num = 3;%µü´ú´ÎÊý

%%³õÖµ;
x10 = 0;
x20 = 0;
y0 = 0;

%%
m1 = 2*k1*v1/s;
m2 = k/s*(v0+v1*exp(-k1*rou1^2));


%Df = df1/dx1£¨Æ«µ¼£©
Df = @(x1)m1.*(x1./s-rou1).*exp(-k1.*(x1./s-rou1).^2) -
m2.*exp(-k./s.*x1);
figure(3);
plot(0:100,Df(0:100),'k:');xlabel('x1'); ylabel('Df');
title('df1/dx1');

%%
    function dx1 = Dx1DtSubFun(t, x1)
        dx1 =
-v0-v1*exp(-k1*(x1/s-rou1).^2)+(v0+v1*exp(-k1*rou1^2))*exp(-k*x1/s);
        dy = zeros(size(t));
        if length(t)>1
            for j = 1:length(t)
                if t(j) < 1
                    dy(j) = -(beta(t0)*exp(-t(j)*Df(0.5*t(j))) + c2) /
(2*lambda);
                else
                    dy(j) = -(beta(t0) /beta(t(j))+c2)/(2*lambda);
                end
            end
        else
            if t < 1
                dy = -(beta(t0)*exp(-t*Df(0.5*t)) + c2) / (2*lambda);
            else
%                 if isnan(beta(t)) && ~isnan(beta(t0))
%                     disp(['t:',num2str(t)]);
```

```
%                  disp(['beta(t):',num2str(beta(t)),'
beta(t0):',num2str(beta(t0))]);
%              end
              dy = -(beta(t0) /beta(t)+c2)/(2*lambda);
          end
      end
      dx1 = dx1 + dy;
   end


epsilon = 10^(-3);
beta = @(t)exp(-0.003*t);%²Â²â³õÊ¼Öµ-0.0001 µü´ú5´Î
betafirst = beta;

tspan = [0, t0+10];
x10 = 0;

for kk = 1:2
   for k = 1:Num

   tao = 0:step:t0;
   tmp = 0;
   for i = 1:length(tao)-1
      if tao(i)<1%ÎªÁËÈ¥³ýwarning
         tmp = tmp + step*exp(-tao(i)*Df(0.5*tao(i)));
      else
         tmp = tmp + step*1/beta(tao(i));
      end
   end
   integ = tmp;%»ý·Ö

   c2 = (-beta(t0) * integ - 2*lambda*yt0 )/t0;
   %c2 = c2 - 1

   dy0 = -1/(2*lambda)*(beta(t0)+c2);
   dyt0 = -(1+c2)/(2*lambda);
   dx1t0 = dyt0 - lambda*dyt0^2;


   options = odeset('RelTol', 0.001);
   sol = ode45(@Dx1DtSubFun, tspan, x10, options);
   t45 = sol.x;
   x45 = sol.y;
```

```
    figure(1);clf; h1 = axes; %hold on;
    figure(2);clf; g1 = axes; %hold on;
    plot(h1, t45, x45, 'r:'); xlabel(h1, 't'); ylabel(h1, 'x1'); title(h1,
'x1(t)');
    plot(g1, x45, Dx1DtSubFun(t45, x45), 'b-'); xlabel(g1, 'Î»ÒÆx1');
ylabel(g1, 'ËÙ¶Èdx1'); title(g1, 'x1µÄÏàÆ½Ãæ');


    [x45t0,dx45t0] = deval(sol, t0);

    %%
    dy = @(tt)-(beta(t0)/beta(tt) + c2) / (2*lambda);
    z = 0:step:t0;
    y = zeros(size(z));
    dydt = zeros(size(z));
    for i = 1:length(z)
        if z(i)<.5%ÎªÁËÈ¥³ýwaring
            dydt(i) = -(beta(t0)*exp(-z(i)*Df(0.5*z(i))) + c2) /
(2*lambda);
        else
            dydt(i) = dy(z(i));
        end

        if i==1
            y(1) = 0;
            continue;
        end
        y(i) = y(i-1)+ step*0.5*(dydt(i)+dydt(i-1));%»ý·Ö
    end

    %ÏÔÊ¾y(t)
    figure(4);
    plot(z, y, 'b:');xlabel('t'); ylabel('y'); title('y(t)');
    %ÏÔÊ¾y'(t)
    figure(5);
    plot(z, dydt, 'b:');xlabel('t'); ylabel('dy/dt'); title('dydt');


    %%
    if kk == 1
        dydtsquare = dydt.^2;
        x2t0tmp = -x45t0 + yt0 - lambda * step * sum(dydtsquare(1:end-1)
+ 0.5*diff(dydtsquare))
```

```
        if k == 1
            x2t0 = x2t0tmp;
            bestnum = k;
        end

        if x2t0 < x2t0tmp & x2t0tmp > 0
           x2t0 = x2t0tmp;
           bestnum = k;
        end
    end
    %%
    betatmp = @(t)Df(deval(sol, t));
    beta = @(tt)exp(quadl(betatmp, 0, tt));
    end

    if kk == 1
        Num = bestnum;
        disp(['bestnum:',num2str(bestnum)]);
        beta = betafirst;%²Â²â³õÊ¼Öµ
    end
end

end
```